RELOCATION BATCH PROCESSING FOR DISK DRIVES

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. Provisional Application No. 60/426,721, filed November 15, 2002.

BACKGROUND OF THE INVENTION

[0002] One of the tasks a computer performs when a disk is being formatted is detecting bad sectors on the disk and relocating them to minimize their interference with performance. Such relocations are known as primary or static relocations and are effected by a relocation algorithm called a slipping algorithm. Bad sectors and debris may also be discovered after formatting is complete, causing the sectors to be relocated after the disk is in service. These relocations are called secondary or dynamic relocations. Secondary relocations are typically performed using an algorithm called a replacement algorithm.

[0003] Error information regarding bad sectors encountered during formatting and during subsequent disk usage are typically stored in a specified area on the disk, sometimes called the primary defect list (PDL) and the secondary defect list (SDL) respectively. This information indicates which sectors should not be used to store data, and will cause processing commands (e.g., read, write, verify) to either skip the bad sectors or search other specified physical locations on the disk for those sectors.

[0004] Whereas slipping algorithms have a minimal effect on drive performance, replacement algorithms adversely affect the performance of a disk drive, both during the write process and subsequently during the read process. Dynamic relocations are time consuming due to the excessive seek time they require. During a normal write process, the drive accesses process sectors in the prime or standard user area of the disk. When a bad sector is encountered in the standard user area, the drive seeks to a dedicated defect management area (DMA) and sets up a relocated sector for the data that was to be placed

in the bad sector. The DMA is physically separated from the standard user area. Whenever the bad sector is referenced, the drive is then translated to the appropriate DMA where the relocated sector lies. After the data in the relocated sector has been processed, the drive seeks back to the standard user area and continues.

[0005] The process of seeking to the DMA, processing the relocation, and then seeking back to the standard user area is repeated every time a relocation is needed. Thus, a seek-process-seek delay is encountered every time a relocated sector needs to be accessed. This time penalty grows linearly with the number of defective sectors in an extent.

[0006] Prior efforts at reducing this time penalty have included providing multiple DMA areas on the disk. The computer selects the DMA area closest to the bad sector whenever a relocation must be performed. This method reduces the constant associated with the linear time penalty because the seek step takes less time. However, the linear nature of the time penalty has not been altered.

[0007] The transfer rate of disk drives is severely degraded by relocations due to the linear time penalty associated with the typical defect management process. As data densities increase, the effect of debris and reduction in margins will increase the number of relocations for disk drive systems.

[0008] There is a need for a process that reduces the time required to conduct a post-formatting relocation. There is also a need for a process that allows the seek steps of several relocations to be combined, thereby encouraging batch relocations.

BRIEF SUMMARY OF THE INVENTION

[0009] The present invention generally pertains to a process for efficiently performing multiple post-formatting relocations of bad disk sectors by allowing multiple relocations to be performed simultaneously. The amount of debris or other physical flaws present on removable media such as disk 50 typically increases over time. Further, unlike defective

sectors that are identified during formatting, these post-formatting defects are not managed using the sector slipping algorithm, and it is time consuming to address such flaws. The time required for managing these secondary/dynamic relocations are minimized in accordance with the present invention.

[0010]The process of the present invention involves processing (writing, reading, verifying, etc.) the data in the prime area of the disk. As bad sectors are encountered during the processing, the relocations are saved collectively in memory for later processing. The data processing continues in this manner until completion is achieved. Upon completion of the data processing in the prime area, a single seek step is used to access a designated DMA. All of the relocations are then processed in the DMA.

[0011]Accordingly, the present invention provides for completing all of the prime disk area processing first and only then batch processing the secondary relocations. This process avoids implementing a lengthy linear replacement algorithm to address the post-formatting defects, which would otherwise call multiple seek operations for each defective sector identified on disk 50. The approach of the present invention reduces the linear time penalty in the number of relocations to a near constant time.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012]Figure 1 is a diagram of a data storage system in accordance with the present invention;

[0013]Figure 2 is a diagram showing the mapping of logical sectors to physical sectors;

[0014]Figure 3a is a diagram illustrating the disk layout after formatting but before defective sector relocations have been processed; and

[0015]Figure 3b is a diagram again illustrating the disk layout, however now shown after defective sector relocations have been processed.

DETAILED DESCRIPTION OF THE INVENTION

[0016] Referring now to Figure 1, there is shown a data storage system 10 including a disk controller 30 and a disk drive 40. Disk controller 30 receives data requests from a host computer 20, which causes disk controller 30 to communicate with disk drive 40. Data storage system 10 further includes a random access memory (RAM) 60, which is in communication with disk controller 30. Disk drive 40 houses a disk 50, such as a magnetic or optical disk or other removable media, and reads data from and writes data to the disk. Disk drive 40 also includes a number of control systems and components which are obviously not shown in Figure 1. Disk 50 includes numerous sectors (not shown) for storing data, which are the smallest units that can be accessed on a disk. Disk 50 may contain one or more bad sectors, which is a sector that cannot be used due to a physical flaw. Disk drive 40 and disk controller 30 operate to control disk drive 40 for accomplishing the various tasks that are performed on the disk 50 as described herein.

[0017] During disk formatting, in accordance with the present invention, the primary relocation, also known as static relocation, erases the data in the sectors of disk 50 and checks the surface of disk 50 for any defective sectors. When bad sectors are identified, their physical locations are stored by data storage system 10 in RAM 60, and the physical/logical sector relationship is mapped out using a sector slipping algorithm. When disk 50 is new and the formatting process identifies bad sectors, such sectors are typically truly defective and are therefore unusable for the life of the disk. A sector slipping algorithm provides for a relatively fast means of identifying and managing bad sectors and debris found during formatting, thereby minimizing the disk access time required by disk drive 40 to format disk 50.

[0018] Figure 2 shows an example of the relationship between the logical sectors and physical sectors of disk 50, where the disk has 1000 physical sectors, sectors 0-999. If during the process of formatting disk 50 it is determined by the data storage system 10 that physical sectors 4 and 7 of disk 50 are defective, the primary/static replacement procedure maps the logical sectors accordingly. As shown in Figure 2, physical sectors 4 and 7 have

no logical sectors mapped thereto; the mapping of the logical sectors proceeds sequentially but passes over physical sectors 4 and 7, which causes those sectors to be skipped during disk usage.   The disk then has only 998 physical sectors available for subsequent processing after formatting.

[0019]As indicated, the data storage system uses a sector slipping algorithm during formatting, which causes disk drive 40 to merely skip over defective sectors when there is an attempt to access them.   Thus, the time associated with the formatting process is minimized due to the short period needed to skip a disk sector, which is typically the time necessary to read a sector.

[0020]Once disk 50 has been formatted, the disk is available for standard processing in the prime disk area.   Standard processing typically includes activities such as writing to the disk, reading from the disk, or verifying data on the disk.   When defective sectors and debris are encountered during the standard processing, secondary sector relocations are implemented.   Secondary sector relocations adjust logical sectors that are mapped to defective physical sectors in a Prime Disk Area (PDA) of disk 50 and maps them to good physical sectors in a defect management area (DMA) of disk 50.   Figure 3a illustrates the PDA and the DMA of disk 50.

[0021]Figure 3a shows a post-formatted disk 50 in accordance with the present invention, where the disk has 1000 physical sectors and none of the physical sectors have yet been determined to be defective.   Figure 3a further illustrates the relationship between logical sectors and physical sectors; the PDA has 990 physical sectors, which are currently mapped to by logical sectors 0-989, and the DMA has 10 physical sectors, which are not yet mapped.   The ten physical sectors in the DMA are reserved for subsequent mapping when defects are encountered.

[0022]In operation, all standard data processing (e.g., reads, writes, verifies) in the PDA of disk 50 is performed first without processing any relocations in the DMA of disk 50. Accordingly, during the standard data processing, disk drive 40 does not access the DMA, even if defects are encountered.   Rather, as defective sectors are encountered by disk

drive 40 during the initial standard data processing, information regarding those sectors is simultaneously stored in RAM 60 for subsequent processing. Because disk drive 40 does not access the DMA of disk 50 for each defective sector encountered, the access times involved with this first step are comparable to the initial formatting, which is typically the same time necessary to read/write/verify a sector, depending on the operation being performed.

[0023] Next, once the initial standard data processing of the prime disk area is complete, all secondary relocations are batch processed. In order to perform the secondary relocations, disk drive 40 executes only a single seek-process, which seeks to the DMA of disk 50, and then performs the necessary relocation batch processing for multiple defective sectors. The batch processing is accomplished by disk drive 40 accessing RAM 60, retrieving the necessary defective sector information stored therein, and then executing the relocations based on that information.

[0024] By way of example, in accordance with the present invention post-formatting disk usage may manage subsequent defects as shown in Figures 3a and 3b. As previously described, Figure 3a shows disk 50, which has 1000 physical sectors: 990 sectors (numbered 0-989) designated as the PDA and 10 sectors (990-999) designated as the DMA. When post-formatting processing (e.g., a write command) occurs to a set of logical sectors, such as logical sectors 0-10, disk drive 40 first attempts to write to the correspondingly mapped physical sectors in the PDA, as shown in Figure 3a. If any of the physical sectors of disk 50 mapped to by logical sectors 0-10 are defective or debris is encountered at those sectors, the defect information is stored in RAM 60 for subsequent processing, and disk drive 40 continues writing to the PDA until all of the corresponding physical sectors of disk 50 have been accessed. For example, if during the write command disk drive 40 attempts to access the physical sectors in the PDA that are mapped to logical sectors 0-10, logical sectors 4 and 9 may be identified as defective. If those physical sectors in the PDA are identified as defective, the data storage system 10 stores the appropriate defect information in RAM 60.

[0025] Once the write command to sectors 0-10 in the PDA is otherwise completed, disk drive 40 then seeks once to the DMA of disk 50 and processes the relocation of the physical sectors that were identified as defective during execution of the write command. The data storage system 10 processes the relocation of those sectors only after disk drive 40 writes or at least attempts to write to each of the physical sectors in the PDA. In this example, after the standard processing completes the write command, logical sectors 4 and 9 are re-mapped in batch to the DMA based on the defect information in RAM 60; disk drive 40 seeks to the DMA and then writes the logical sector 4 and 9 data to the corresponding physical sectors in the DMA, as shown in Figure 3b. As the process of the present invention is implemented on disks of increasing density and the number of defective sectors increases, the more the relocation time penalty will be reduced.

[0026] Those skilled in the art will further appreciate that the present invention may be embodied in other specific forms without departing from the spirit or central attributes thereof. In that the foregoing description of the present invention discloses only exemplary embodiments thereof, it is to be understood that other variations are contemplated as being within the scope of the present invention. Accordingly, the present invention is not limited in the particular embodiments which have been described in detail therein. Rather, reference should be made to the appended claims as indicative of the scope and content of the present invention.